

REMARKS

Claims 1-22 are currently pending in this application. In an Office Action dated September 21, 2005 (hereinafter the "Office Action"), Claims 1-3, 9-13, 16-18, 21, and 22 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,457,065, issued to Rich et al. (hereinafter "Rich et al."). Claims 4-8, 14-15, 19, and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Rich et al. in view of U.S. Patent No. 6,108,664, issued to Nori et al. (hereinafter "Nori et al."). Applicants respectfully request reconsideration and allowance of this application for the reasons set forth below.

Prior to discussing in detail why applicants believe that all of the claims in the application are allowable, a brief description of applicants' invention and the cited references is provided. The following discussions of the disclosed embodiments of applicants' invention and the teachings of the applied references are not provided to define the scope or interpretation of any of applicants' claims. Instead, such discussed differences are provided to help the U.S. Patent and Trademark Office (hereinafter "the Office") better appreciate important claim distinctions discussed thereafter.

Summary of the Invention

In accordance with the present invention, a system and method for transmitting data between mirrored databases stored on both client and server computers is provided. More specifically, the present invention provides a system and method for efficiently synchronizing multiple databases and database items in those databases while maintaining parent-child relationships between related database items. The method and system allow a client computer to update a server database while minimizing the number of database items transmitted between the client and server computers. By minimizing the transmission of data, the system and method support more efficient communication between computers.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

The present invention utilizes the parent-child relationships that exist between database items to determine if certain child database items stored in the client computer should be communicated to the server computer. According to one embodiment, when the client database is synchronized with the server database, parent database items are first communicated from the client to the server. Next, child database items associated with parent database items are communicated if there was not a communication error in the transfer of the parent database items. The method does not communicate a child database item if its associated parent database item is not properly communicated. Once the set of database items has been communicated, the database items contained on the server computer are synchronized with the client computer. If any particular database item is not successfully uploaded from the client computer to the server computer, lower level database items, i.e., children, grandchildren, etc., of the particular database item are all bypassed in the synchronization process and not uploaded to the server computer. In addition to providing an efficient synchronization method, this embodiment allows the server and client to generate database item IDs in duplicated databases without generating conflicting ID numbers for identical database items.

Summary of Rich et al.

Rich et al. purportedly discloses a processing system for improving the performance of distributed object systems. As is known to those skilled in the art and others, object-oriented programming is a type of programming in which developers define data types and operations (i.e., functions or methods) that may be applied to a data type. Rich et al. purportedly discloses a system where users may modify "objects" from an object-oriented programming language on remote computer systems and replicate changes to the objects in a global repository. When a user of a workstation accesses an object stored in a global repository, the object is replicated to the user's workstation. Thereafter, modifications to the object occur locally at the workstation. When the user requests to commit the modifications, a determination is first made as to whether

committing the modifications will result in an unacceptable data conflict. If no unacceptable data conflict occurs, the modifications are committed to the object stored in the global repository. Rich et al. does not disclose or suggest a method that selectively communicates database items to synchronize based on the transmission of a successor (i.e., parent, grandparent, etc.) database item.

Summary of Nori et al.

Nori et al. purportedly discloses a system and method for presenting and modifying data from a set of tables in a database. Nori et al. discloses general methods for defining views based on a set of tables that includes relational tables or object tables. The view defines the presentation of data from the one or more tables as a set of objects that resides in the database. Data is read from the rows of the tables based on the view and is presented as a set of objects that resides in the database. An object ID that is based on data from the one or more rows is generated and associated with each object presented. Nori et al. Abstract. Nori et al. also discloses a generally known method for assigning database IDs to objects when an object is created. *Id.* at Col. 8, lines 53-60.

Rejection of Claims 1-3, 9-13, 16, and 18 Under 35 U.S.C. § 102

The Office Action rejected Claims 1-3, 9-13, and 18 under 35 U.S.C. § 102(e) as being anticipated by Rich et al.. The Office Action asserts that Rich et al. discloses each and every element of these claims. As described in more detail below, applicants respectfully disagree.

Claim 1 recites:

A computer-implemented method of optimizing the synchronization of data between a client computer having a client database and a server computer having a server database, comprising:

communicating a plurality of database items that maintain a parent-child hierarchical tree relationship from the client computer to the server computer, wherein the plurality of database items includes a parent database item;

determining if the parent database item is improperly received, wherein the determination of an improperly received parent database item is based upon the detection of a data transfer error that is characteristic of an error condition associated with the parent database item;

selectively communicating a child database item associated with the parent database item from the client computer to the server computer, only if a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item; and

synchronizing the communicated child database item with a corresponding child database item stored on the server computer.

Claim 1 of the present invention defines a method where data communicated between server and client computers includes "communicating a plurality of database items that **maintain a parent-child hierarchical tree relationship** from the client computer to the server computer, wherein the plurality of database items includes a parent database item." [Emphasis added.] Conversely, objects as disclosed in Rich et al. do not maintain a hierarchical tree relationship. The Office Action states that Rich et al. discloses a system whereby database items are maintained in a hierarchical tree structure similar to database items disclosed in the present application. More specifically, the Office Action states that Rich et al. discloses a distributed computer system that forms "networks of nodes where each node may act as both a client and a server to the other nodes and data objects include parent/child/grandchild or successor objects." Office Action at page 3. In support of that proposition, the Office Action cites Rich et al. at Col. 8, lines 63-65, and Col. 9, lines 5-67. However, the referenced sections of Rich et al. do not disclose a system whereby database items are communicated between computers that maintain a hierarchical tree relationship. Moreover, applicants are unable to find any reference in Rich et al. to communicating database items that includes a parent database item.

Applicants acknowledge that Rich et al. mentions hierarchical parent/child tree structures in various aspects which are commonly used in the art in various respects. However, applicants submit that Rich et al. does not disclose the combination of selectively communicating "database

items that maintain a parent-child hierarchical tree relationship." The relevant section of Rich et al. referenced in the Office Action states:

[a] physical node may have nodes of transaction trees for multiple transactions and may be a parent for some transactions while being a child for other transactions. The present invention uses these nested transaction trees to manage changes to the replicas, as well as actual changes to the remote objects in the persistent data store, resulting from one or more concurrent and/or nested transactions.

Rich et al. at Col. 9, lines 5-12.

Rich et al. purportedly discloses "transaction trees" that may become nested in order to manage actual changes to remote objects. Stated differently, "changes made to a replicated object by a transaction are represented in a tree structure that is internally managed by the application." Rich et al. Abstract. As described in Rich et al., when changes to an object will be committed to the persistent data store, a transaction tree may be transmitted to the computer that maintains the persistent data store. Based on the data in a transaction tree, the original object is modified using the transaction tree that represents changes made on a remote computer. However, the referenced sections of Rich et al. make it clear that the "objects" disclosed in Rich et al., that the Office Action equates with the "database items" in the present application, do not maintain a hierarchical parent/child relationship. Moreover, the referenced sections of Rich et al. make it clear data is not transmitted between computers based on whether a related database item was successfully transmitted. Instead, Rich et al. discloses a system where potential modifications to an object are represented in a tree data structure in which nodes of the tree maintain a hierarchical relationship. The entire transaction tree disclosed in Rich et al. is transmitted between computers so that modifications that are made may be committed to a global repository. Otherwise, the changes made on the remote computer to an object that is stored in a global repository would not be committed. By contrast, the present invention does not transmit data that represents modifications made to database items between computers. Instead, the

present invention provides logic for determining whether a database item, in its entirety, is transmitted between computers. More specifically, the present invention selectively communicates "a plurality of database items that maintain a parent-child hierarchical tree relationship," which is not equivalent to transmitting "transaction trees" that describes modification made to an object.

The Office Action states that Rich et al. discloses the combination of "selectively communicating a child database item associated with the parent database item from the client computer to the server computer, only if a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item." In support of that proposition, the Office Action cites Rich et al. at Abstract; Figures 4A and 4B; Col. 4, lines 55-64; Col. 10, lines 11-41; and Col. 11, lines 20-67. However, the referenced sections of Rich et al. do not disclose the combination of **"selectively communicating a child database item associated with the parent database item from the client computer to the server computer, only if a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item."** Specifically Rich et al. states:

When an application or application user has made modifications to a replicated object and requests to commit the modifications, a determination is first made as to whether committing the modifications will result in an **unacceptable data conflict**.

[Emphasis added.] Rich et al. Abstract.

Simply stated, a "data transfer error" as recited in Claim 1 is not equivalent to an "unacceptable data conflict" as disclosed in Rich et al.. One benefit to structuring database items in a parent-child hierarchical tree relationship is that the amount of data communicated between computers may be reduced. More specifically, child database items are only transmitted between computers "if a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item." Stated differently, the synchronization of a child

database item depends on the result of the synchronization of a parent database item. Logic for "selectively communicating" a database item based on the relationship with another database item as disclosed in the present invention reduces the amount of data communicated between computers that maintain synchronized databases.

In contrast, the decision to commit modifications to a remote persistent object store only if an "unacceptable data conflict" will not occur is not the same as a system that selectively communicates database items only if a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item. Rich et al. purportedly teaches a system of improving the performance of the distributed object systems. As described in Rich et al., objects are used in a distributed object system which "is an execution environment in which objects may be located on more than one physical computer and the application programs which operate upon those objects may also be located on (and operating on) more than one computer. . . ." Rich et al. at Col. 1, lines 27-30. Examples of distributed object systems provided in Rich et al. include Enterprise JavaBeans ("EJB's) and the Common Object Request Broker Architecture ("COBRA"). *Id.* at Col 1, lines 40-45. Moreover, those skilled in the art and others will recognize that in distributed object systems a client computer performs actions on an object through a well-defined interface that is typically described in an interface definition language. In this regard, a client may request an object's instantiation, perform method invocation on the basis of a specific object instance, receive a result in a response, and the like. As disclosed in Rich et al., objects are replicated on client-based computers in the distributed system from a server-based computer. Thereafter, "method invocations on the object occur locally, avoiding the performance overhead of frequent round trips to the remote persistent object store." Rich et al. Abstract. Changes made to a replicated object are represented using a tree structure. When an application makes changes to a replicated object and requests to commit the modifications to the persistent object store, a determination is first made as to whether

committing the modifications will result in a "unacceptable data conflict." An unacceptable data conflict may occur in this type of situation if objects that are modified on different computers contain conflicting data. These types of data conflicts are expected in a distributed execution environment in which multiple client-based computers may modify objects before submitting the objects to a remote persistent object store. For example, multiple client computers may have the same object replicated on each client computer. Thereafter, changes may be made to the object on each computer that may or may not be compatible. As stated in Rich et al. the process of resolving a data conflict comprises:

determining that no unresolvable data conflict will occur if one of an object version from a child transaction and a parent transaction which are to be merged has not been modified or where both of the object versions have been modified from the child transaction and the parent transaction, logic is applied that determines that the object versions can be successfully merged; and wherein performing the merging further comprises involving application-specific logic to resolve any resolvable data conflicts.

Rich et al. at Col. 4, lines 55-64.

By contrast, a computer that implements aspects of the present invention may determine that a data transfer error that is characteristic of an error condition was not detected with respect to the parent database item "if the date of a record is invalid, or if the server computer 82 determines that certain object account information is invalid, such as a user password. Application at page 21, lines 14-17. A data transfer error that is characteristic of an error condition signifies that a problem occurred that would not be expected during the transfer of data. When the data transfer error is encountered, the present invention selectively communicates database items and thereby minimizing the communication of unnecessary data. Thus, applicants submit that a system that selectively communicates database items based on the occurrence of data transfer error that is characteristic of an error condition is different and is not disclosed by the Rich et al. system that attempts to commit and merge objects that are maintained

on a plurality of computers that may cause data conflicts to occur in the typical course of operations.

The distinction between the synchronization of "database items" as recited in Claim 1 of the present invention and the merging and/or committing "objects" in a repository as purportedly disclosed in Rich et al. is substantive as "database items" are fundamentally different and used in different ways than "objects." Objects replicated in Rich et al. are fundamentally different from the database items replicated by the method recited in Claim 1. More specifically, objects as disclosed in Rich et al. are not database items that are maintained in "a client database" or "a server database." Instead, objects disclosed in Rich et al. operate in "an execution environment" (*Id.* at Col. 1, lines 27-30) in which "method invocations on the objects occur locally." *Id.* at Abstract. Objects that operate in an "execution environment" such as a distributed object system and consist of a different set of information and attributes than database items that operate in and among databases. These objects are the type that are used in object-oriented programming. The Office Action states that "Rich et al. teaches a method of optimizing the synchronization of data between a client computer having a client database and a server computer having a server database. That is, optimization for replication *of objects* between a client node having client database and server node having server database" [Emphasis added.] Office Action at page 9. Claim 1 of the present invention defines a method where the type of data communicated between computers is not an "object" as purportedly disclosed in Rich et al.. More specifically, Claim 1 of the present invention recites "communicating a plurality of database items" between a server computer and a client computer. Simply stated, the method disclosed in Rich et al. does not communicate database items as recited in Claim 1. Instead, as stated in the Office Action, Rich et al. purportedly discloses a system of committing and/or merging "objects" of the type used in object-oriented environments.

Applicants respectfully submit that Rich et al. fails to expressly or inherently teach, disclose, or suggest each and every element of Claim 1. For instance, as explained above, Rich et al. fails to disclose or suggest "selectively communicating a child database item associated with the parent database item from the client computer to the server computer, only if no data transfer error was detected that is characteristic of an error condition with respect to the parent database item." Accordingly, for at least these reasons, applicants respectfully submit that the rejection of Claim 1 is in error and request that it be withdrawn.

Claim 2 is directed to a synchronization method where the communication of a child database item depends on the communication of a parent database item. More specifically, Claim 2 recites:

A method of optimizing the synchronization of data between a client computer having a client database and a server computer having a server database, wherein each database comprises a plurality of database items arranged in a hierarchy comprising at least one parent database item and at least one associated child database item, comprising:

receiving at the server computer a parent database item transferred from the client computer that maintains a parent-child hierarchical tree relationship with child database items in the client database;

assigning a status code to the parent database item received at the server computer, the status code being based upon the detection or non-detection of a data transfer error that is characteristic of an error condition;

transmitting the status code assigned to the parent database item to the client computer, wherein the status code is assigned to the parent database item stored in the client database;

updating a status code of child database items associated with the parent database item in the client database, the updated status code of child database items being based on the status code of the parent database item;

selectively communicating child database items associated with the parent database items from the client computer to the server computer, wherein child database items associated with the parent database item are communicated if the status code indicates a non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item; and

synchronizing the communicated child database item with a corresponding child database item stored on the server computer.

As noted above, Rich et al. does not disclose or suggest any method where the communication of one database item, such as a child database item, depends on the communication of another database item, such as a parent database item. Moreover, Rich et al. does not disclose a method that synchronizes child database items while maintaining a parent-child hierarchical structure between the items. Lastly, Rich et al. does not disclose "assigning a status code to the parent database item received at the server computer, the status code being based upon the detection or non-detection of a data transfer error." The Office Action states that Rich et al. discloses a system of assigning a status code to the parent database item received at the server computer, the status code being based upon the detection or non-detection of a data transfer error. In support of that proposition, the Office Action cites Rich et al. at Col. 4, lines 55-64; Col. 12, lines 33-41; and Col. 16, lines 56-65. However, the referenced sections of Rich et al. do not disclose a system whereby a status code is assigned based on the detection or non-detection of a data transfer error that is characteristic of an error condition. Specifically, the relevant portion of Rich et al. states:

Accesses to a version of an object preferably set (as appropriate) the "modified" flag, described earlier **when changes are made to the version**, and increment the modification level associated with the version. In this manner, the merge and commit procedures are optimized. (For example, by using the external synchronization flag, objects that have been accessed only in read mode do not need to be committed to the persistent store, thereby increasing the efficiency of the commit procedure).

[Emphasis added.] Rich et al. at Col. 16, lines 56-65.

Applicants respectfully submit that assigning a status code to a database item "based upon the detection or non-detection of a data transfer error" is not the as same as setting a modified flag based on "when changes are made" to an object. Therefore, since Rich et al. fails

to expressly or inherently teach, disclose, or suggest each and every element of Claim 2, applicants respectfully submit that the rejection of Claim 2 is in error and request that it be withdrawn.

Since Claims 3, 9, and 10 depend from Claim 2 and Claims 11 and 12 are computer-readable medium and apparatus claims having language that parallels the language of Claim 2, the analysis applied to Claim 2 also applies to these claims. Therefore, applicants respectfully submit that Claims 3 and 9-12 are in condition for allowance for the same reasons as Claim 2. In addition, applicants submit that the dependent claims are allowable for additional reasons described below.

Claim 3 recites a method where a status code is assigned by a server computer. The Office Action states that Rich et al. discloses a system where a status code is assigned by the server computer. In support of that proposition, the Office Action cites Rich et al. at Col. 12, lines 20-26. However, the referenced sections of Rich et al. do not disclose a system whereby a status code is assigned by the server computer. Specifically, the referenced section of Rich et al. states:

As in the child node, the new version is based on the server transaction's parent. Once the new version has been registered into the server transaction, the version is serialized and replicated to the child node. In the child node, the version is deserialized and registered into the current client transaction and returned to the application.

Rich et al. at Col. 12, lines 20-26.

The cited section of Rich et al. discloses a system where objects are reregistered and replicated. However, applicants are unable to find any reference in Rich et al. to a server computer assigning a status code to a database item. Thus, for at least these reasons, applicants respectfully submit that the rejection of Claim 3 is in error and request that it be withdrawn.

Claim 13 recites a synchronization method where the communication of a child database item depends on the communication of a parent database item. In this method, a parent database

item is transmitted from a client computer to a server computer. The method also includes "selectively communicating child database items associated with the parent database item from the client computer to the server computer, wherein child database items associated with the parent database items are communicated if the status code indicates a non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item."

As noted above, Rich et al. does not disclose or suggest any method where the communication of one database item, such as a child database item, depends on the communication of another database item, such as a parent database item. Also, objects replicated in Rich et al. are fundamentally different from the parent and child "database items" recited in the present claims. Therefore, Rich et al. fails to disclose a method of "selectively communicating child database items associated with the parent database items from the client computer to the server computer, wherein child database items associated with the parent database item are communicated if the status code indicates a non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item." Also, Rich et al. does not disclose applicants' claimed combination of steps utilizing mapping information. Mapping information is simply absent from this cited reference. Thus, for at least these reasons, applicants respectfully submit that Claim 13 and its respective dependent claims are in condition for allowance.

Claim 18 is directed to a synchronization method where the communication of a child database item depends on the communication of a parent database item. In this method, a parent database item is received at a server computer. The method also includes:

assigning a status code to the parent database item, the status code being based upon the detection or non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item

and:

transmitting the status code assigned to the parent database item from the server computer to the client computer, wherein the status code enables the client computer to selectively communicate at least one child database item associated with the parent database item if the status code assigned to the parent database item indicates the absence of a data transfer error; and receiving at the server computer at least one child database item associated with the parent database item, if the status code assigned to the parent database item indicates the absence of a data transfer error.

As noted above, Rich et al. does not disclose or suggest any method where the communication of one database item, such as a child database item, depends on the communication of another database item, such as a parent database item. Accordingly, applicants submit that Rich et al. does not disclose a method that includes "receiving at the server computer at least one child database item associated with the parent database item, if the status code assigned to the parent object indicates the absence of a data transfer error that is characteristic of an error condition." Thus, for at least these reasons, applicants submit that Claim 18 and its respective dependent claims are in condition for allowance.

Rejection of Claims 4-8, 14-15, and 19-20 Under 35 U.S.C. § 103(a)

The Office Action rejected Claims 4-8, 14-15, and 19-20 under 35 U.S.C. § 103(a) as being unpatentable over Rich et al. in view of Nori et al.. The Office Action asserts that Rich et al. and Nori et al. suggest each and every element of these claims and that it would be obvious to combine the teachings of Rich et al. and Nori et al.. Applicants respectfully disagree.

As noted in detail above, the independent claims from which Claims 4-8, 14-15, and 19-20 depend, Claims 2, 13, and 18, respectively, are all directed to methods where the communication of one database item, such as a child database item, depends on the communication of another database item, such as a parent database item. Moreover, objects replicated in Rich et al. are fundamentally different from the parent and child "database items"

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

replicated in the present invention. As described below, applicants respectfully submit that Rich et al., alone or in combination with Nori et al., fails to disclose or suggest this and other claimed features.

Nori et al. purportedly discloses a system and method for presenting and modifying data from a set of tables in a database. Nori et al. discloses a general method for defining views based on a set of tables that include relational tables or object tables. Nori et al. also discloses a generally known method for assigning database IDs to objects when an object is created. *Id.* at Col. 8, lines 53-60. There is no suggestion of a method that synchronizes database items mirrored in multiple databases. In addition, Nori et al. does not make any reference to or suggest combining its disclosed methods with a synchronization method.

With respect to Claims 4-8, referring to Claim 2, applicants submit that Rich et al., alone or in combination with Nori et al., fails to disclose or suggest the claimed combination of steps, including the combination of (1) receiving at the server computer a parent database item transferred from the client computer; (2) selectively communicating child database items associated with the parent database items from the client computer to the server computer, wherein child database items associated with the parent database item are communicated if the status code indicates a non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item; and (3) synchronizing the communicated child database item with a corresponding child database item stored on the server computer.

Although Nori et al. refers to database IDs assigned by the server, the combination of Nori et al. and Rich et al. does not suggest a method including "selectively communicating child database items associated with the parent database items from the client computer to the server computer, wherein child database items associated with the parent database item are communicated if the status code indicates a non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item," and "synchronizing

the communicated child database item with a corresponding child database item stored on the server computer." Moreover, Rich et al. does not make any reference or suggest combining its disclosed methods with an object identification system as disclosed in Nori et al..

In addition, specific to Claim 4, Rich et al. and Nori et al. fail to disclose or suggest other claim elements. Claim 4 defines a claimed combination of steps, including the combination of "assigning a server ID to the database item received at the server computer if no transfer error was detected; and transmitting the server ID assigned to the database item to the client computer." Applicants submit that the Office Action has failed to show, and applicants are unable to find, where Rich et al. discloses a method where an ID is assigned "if no transfer error was detected that is characteristic of an error condition." The Office Action states that "Rich et al. does suggest the server assigns a status code such as a version status of the objects." However, there is no suggestion of a method where server IDs are assigned if no transfer error is detected that is characteristic of an error condition. Stated differently, Rich et al. does not disclose the process of when to assign IDs to database items that are being synchronized between remote computers. Although Nori et al. discloses a method of assigning an ID to a newly-created database object, Nori et al. does not suggest a method of "assigning a server ID to the database item received at the server computer if no transfer error that is characteristic of an error condition was detected; and transmitting the server ID assigned to the database item to the client computer." In view of the foregoing, and by virtue of their dependency on Claim 2, applicants submit that Claims 4-8 are in condition for allowance.

With respect to Claims 14 and 15, applicants respectfully submit that Rich et al., alone or in combination with Nori et al., fails to disclose or suggest the claimed combination, including the Claim 13 recitation of "selectively transmitting child database items from the client computer to the server computer, wherein child database items associated with the parent database item are communicated if the mapping information indicates a non-detection of a data transfer error that

is characteristic of an error condition associated with the parent database item." In addition, Rich et al. and Nori et al. are silent with respect to the use of mapping information to indicate a non-detection of a data transfer error of the parent database item. Since Rich et al., alone or in combination with Nori et al., fail to disclose these features, applicants submit that Claims 14 and 15 are in condition for allowance.

With respect to Claims 19 and 20, applicants respectfully submit that Rich et al., alone or in combination with Nori et al., fails to disclose or suggest the claimed combination, including the independent Claim 18 recitation of "assigning a status code to the parent database item, the status code being based upon the detection or non-detection of a data transfer error that is characteristic of an error condition associated with the parent database item" and "transmitting the status code assigned to the parent database item from the server computer to the client computer, wherein the status code enables the client computer to selectively communicate at least one child database item associated with the parent database item if the status code assigned to the parent database item indicates the absence of a data transfer error; and receiving at the server computer at least one child database item associated with the parent database item, if the status code assigned to the parent database item indicates the absence of a data transfer error."

Even when combined with Nori et al., Rich et al. does not disclose or suggest any method where the communication of one database item, such as a child database item, depends on the communication of another database item, such as a parent database item. Hence, it is clear that Rich et al. and Nori et al. fail to suggest each claim element of Claims 19 and 20. To establish a *prima facie* case of obviousness, M.P.E.P. § 2143 requires that the prior art references "must teach or suggest all of the claim limitations" and that there "must be some suggestion or motivation, either in the references themselves or in knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings." M.P.E.P. § 2143 (February 2003). As described above, the Office Action has failed to show, and

applicants are unable to find, where Rich et al. and Nori et al., alone or in combination, disclose or suggest each and every element of Claims 4-8, 14-15, 19, and 20. Thus, for at least these reasons, applicants respectfully submit that the Office has not established a *prima facie* case of obviousness and respectfully submit that these claims are in condition for allowance.

Rejection of Claims 16-17 and 21-22

Since Claims 16-17 are computer-readable medium and apparatus claims having language that parallels the language in Claims 13-15, the analysis applied to Claims 13-15 also applies to these claims. Similarly, since Claims 21-22 are computer-readable medium and apparatus claims having language that parallels the language in Claims 18-20, the analysis applied to Claims 18-20 also applies to these claims.

CONCLUSION

In view of the foregoing remarks, it is submitted that the present application is now in condition for allowance. Reconsideration and reexamination of the application and allowance of the claims are solicited. If the Examiner has any questions or comments concerning this matter, the Examiner is invited to contact applicants' undersigned attorney at the number below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}



Clint J. Feekes

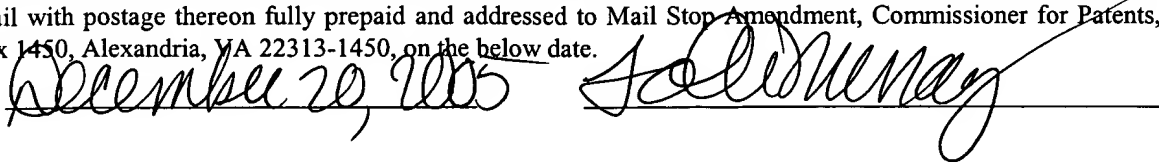
Registration No. 51,670

Direct Dial No. 206.695.1633

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date:

CJF:tmm



LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100